Deciphering Compatibility Relationships with Textual Descriptions via Extraction and Explanation

Yu Wang, Zexue He, Zhankui He, Hao Xu, Julian McAuley

University of California, San Diego
{yuw164, zehe, zhh004, hax019, jmcauley}@ucsd.edu

Abstract

Understanding and accurately explaining compatibility relationships between fashion items is a challenging problem in the burgeoning domain of AI-driven outfit recommendations. Present models, while making strides in this area, still occasionally fall short, offering explanations that can be elementary and repetitive. This work aims to address these shortcomings by introducing the Pair Fashion Explanation (PFE) dataset, a unique resource that has been curated to illuminate these compatibility relationships. Furthermore, we propose an innovative two stage pipeline model that leverages this dataset. This fine-tuning allows the model to generate explanations that convey the compatibility relationships between items. Our experiments showcase the model's potential in crafting descriptions that are knowledgeable, aligned with ground-truth matching correlations, and that produce understandable and informative descriptions, as assessed by both automatic metrics and human evaluation. Our code and data are released at https://github.com/wangyu-ustc/PairFashionExplanation.

Introduction

Fashion and technology now intersect through outfit recommendation platforms, which provide a vast array of apparel and accessory options. These platforms are prevalent, such as Amazon Fashion and Chictopia An essential component of these recommendation systems is explanation capabilities, which guide user decisions and enable system diagnosis. Additionally, as outfit recommendation becomes popular, the need for explanatory functionality extends beyond single items, emphasizing the importance of illustrating compatibility relationships between pairs of items. This presents the central challenge that our research seeks to address: **Given a pair of fashion items, how can we effectively uncover and articulate their intrinsic compatibility relationships?** In order to ensure these explanations are comprehensible to a broad spectrum of users, we prioritize generating explanations as coherent and naturally-phrased sentences.

Existing literature offers a multitude of models for outfit recommendations, which are designed to generate highly compatible outfit combinations (Lu et al. 2021a; Chen et al. 2019). However, these models often overlook the essential component of providing explanations for their predictions. Certain efforts have been made to incorporate explanatory features within prediction models (Wang, Wang, and Su 2022; Papadopoulos et al. 2022), but these attempts frequently result in rudimentary and repetitive explanations, failing to accurately encapsulate the complex compatibility relationships between diverse pairs of clothing items (Lin et al. 2019; Kaicheng, Xingxing, and Wong 2021). This indicates an unresolved challenge in the field: the generation of nuanced and informative explanations that accurately reflect true compatibility relationships.

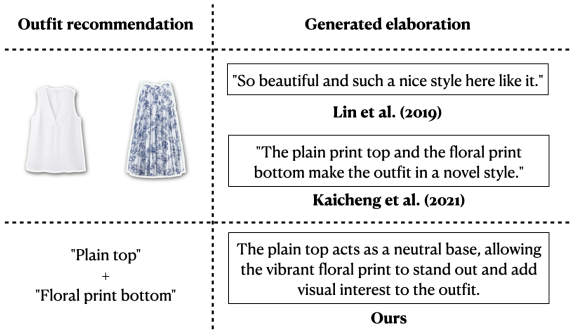


Figure 1: Our task is generating natural language descriptions to explain compatibility relationships between items.

Recent advancements in large language models like ChatGPT and GPT4 (OpenAI 2023) have shown considerable potential across various tasks (Bang et al. 2023). However, their ability to generate expert-aligned explanations for clothing matching remains dubious, with a tendency to produce plausible yet inauthentic explanations that diverge from the true compatibility relationships.

To address this challenge, we present a novel solution combining a newly curated dataset and a two-stage pipeline model. Our **PFE** dataset comprises 6,407 sentences. Each sentence in the dataset serves as a distinct compatibility explanation for a pair of items, providing relevant entities and features. It offers insights into the explanations of clothing compatibility. By focusing on quality rather than quantity, our **PFE** dataset provides dense, high-quality data that can effectively support the intricate task of elucidating clothing

compatibility relationships.

Leveraging the PFE dataset, we further develop a two-stage pipeline that extracts important features from item descriptions and generates corresponding explanations. In the first stage, we use a large general public dataset collected from user interactions to train a feature extractor model. This model extracts important features from the given abundant features with the supervision of positive and negative pairs. The second stage utilizes our PFE dataset to fine-tune a large language model. With the explicit compatibility relationships demonstrated in the dataset, we can inject the domain-specific knowledge into the large language model, enabling the model to discover compatibility relationships between items. During inference, our proposed framework also employs a two-stage approach. We use the first-stage model to extract important features and make predictions. If the prediction is positive, the extracted features are templated as prompts for the second-stage model to generate explanations. This approach ensures that the generated explanations are more likely to make sense and extract the inherent relationships between items. The proposed pipeline allows for a more nuanced and informative approach to generating explanations for item matching.

Our contributions can be summarized as:

1. We curate a novel dataset specifically designed for pair-matching explanations. To the best of our knowledge, this is the first dataset for pairwise explanations. This dataset helps us to fine-tune our model for better performance.
2. We introduce a pipeline that utilizes the proposed dataset and other relevant datasets to extract important features and generate informative and diversified explanations.
3. Experimental results demonstrate that our model generates knowledgeable and aligned descriptions that are more accurate in terms of ground-truth matching correlations compared to existing methods. Meanwhile, our model produces more acceptable and informative descriptions according to human evaluation compared to other recommendation models and general-purpose language models.

Related Work

Outfit Recommendation Within the realm of outfit recommendation, research efforts can broadly be categorized into several strands. A segment of these studies emphasizes recommending comprehensive sets of clothing items, treating outfits as integral units (Chen et al. 2019; Lu et al. 2021b; Guan et al. 2022; Banerjee et al. 2020). Others approach the problem from the perspective of compatibility predictions, generating compatibility scores for individual items within a partial outfit (Sarkar et al. 2023; Song et al. 2023; Zhou, Su, and Wang 2022; Moosaei et al. 2022). Alternatively, there are methods that compute compatibility scores for pairs of items, instrumental in the construction of outfits or item retrieval (Lin, Moosaei, and Yang 2020; Balim and Özkan 2023; Li, Liu, and Forsyth 2019; Castro et al. 2023; Chen et al. 2022). Another line of research evaluates the overall compatibility score of an entire outfit (Wang, Wang, and Su 2022). Certain studies have further explored the concept of incompatibility detection (Papadopoulos et al. 2022; Balim and Özkan 2023), aiming to identify items within an outfit that mismatch. Although these works employ diverse methodologies, a common feature among them is the absence of predictive explanations.

Explanations for Outfit Recommendation While the bulk of outfit recommendation research concentrates on generating recommendations without explanations, a few works have tried to incorporate justifications for their predictions. For instance, Mo et al. (2023) predicts attributes of the missing item, utilizing the predicted attributes as explanations and indicators for item recommendation. In a similar vein, Tangseng and Okatani (2020) detects incompatible items and provides basic reasons for incompatibility, such as mismatched texture or color. Beyond these rudimentary justifications, some research efforts seek to generate natural language explanations. Kaicheng, Xingxing, and Wong (2021) extracts item attributes and identifies those contributing to prediction, subsequently use them to construct explanatory sentences. Lin et al. (2019) also generates textual explanations, although their scope is often limited and lacks detailed insight. In our study, we aim to address these limitations by generating detailed, diverse, and informative explanations for item pair compatibility within outfits.

Methodology

Notation and Problem Formulation

We start by defining an item pair (i, j), where i, j are the indices of the items. Each item i is associated with a distinct set of descriptive words, constructing a bag of words t_i, which describe its color, style, material, and other properties. The features of the item pair (i, j) are denoted as (t_i, t_j). Additionally, we have the category information of the items, which is denoted as (c_i, c_j) for the item pair (i, j). For example, one data instance from FashionVC (Song et al. 2017) for t_i, t_j, c_i, c_j is "Bamford Cropped cashmere cable-knit", "Vince Sequined georgette mini", "sweater", "skirt", respectively.

Given the item pair (i, j) with their corresponding features and categories, our target is to generate the explanation for the pair, which is defined as t_e. To achieve this goal, we propose the extraction model f_θ and the generation model g_φ. The extraction model f_θ extracts the important features in (t_i, t_j):

$$(r_i, r_j) = f_\theta(t_i, t_j) \tag{1}$$

where (r_i, r_j) are the important features in (t_i, t_j) which contribute to whether items (i, j) match or not. While g_φ could generate the explanations to describe why (i_1, i_2) is a matching pair:

$$t_e = g_\varphi(c_i, r_i, c_j, r_j) \tag{2}$$

Here t_e is the explanation for the item pair (i, j).

Stage I: Extract Important Features

In this stage, we require a well-trained feature extraction model f_θ which needs supervision. We use datasets of

matching pairs, constructed as $\{i_{n_1}, i_{n_2}\}_{n=1}^{N}$. To obtain negative pairs, we sample items from the entire set and ensure exclusivity with the positive pairs to form $\{i_{n_1}, i_{n_2}\}_{n=N+1}^{2N}$. The positive and negative pairs provide the supervision to train f_θ. However, since f_θ only performs extraction rather than prediction, we add another model h_ϕ for classification. The training objective then becomes:

$$\min_{\theta} \sum_{n=1}^{2N} \mathcal{L}(h_\phi(f_\theta(t_{i_{n_1}}, t_{i_{n_2}})), y_n) \tag{3}$$

where $(t_{i_{n_1}}, t_{i_{n_2}}))$ is the corresponding feature pair of $(i_{n_1}, i_{n_2}), n \in \{1, \cdots, 2N\}$. And y_n denotes whether the n-th pair is positive ($y_n = 1$) or negative ($y_n = 0$). Once we jointly train f_θ and h_ϕ using the dataset, we obtain a feature extractor that can identify significant features, and h_ϕ can determine whether the pair is a good match. If the pair is deemed a good match, it will proceed to the next stage, where explanations will be generated.

Stage II: Generate Explanations

Next we want to generate reasonable explanations for the item pair (i, j) with the categories (c_i, c_j) and the extracted features (r_i, r_j) in Section . However, supervision in this stage is also missing. So we first construct **PFE**-dataset, where **PFE** stands for Pair Fashion Explanation, which will be used for finetuning our model.

PFE-dataset Construction We manually define a set of key paraphrases: {"Clothes Match", "Clothes Fashion", "Fashion", "Outfit of The Day", "Style", "Match Clothes", "How to Dress", "How to Wear", "Match"}. For each key paraphrase, we conduct a search on public magazines and crawl all related articles. This yields a dataset of 959,157 sentences. However, not all of these sentences are relevant to our task of pair matching. To filter out irrelevant sentences, we adopt the package `spacy` to perform named entity recognition (NER) on the sentence, extracting the entities with the NER tag "Noun". We create a dictionary of relevant entities[1] and extract sentences containing more than two such entities, resulting in a reduced dataset of 48,726 sentences. We then turned to ChatGPT to extract important features from the sentences[3]. The sentences that do not necessarily explain the relationship between items are filtered out. We end up with a final dataset of 6,407 sentences, each with important features, categories, and ground truth explanations denoted by $\{r_i, r_j, c_i, c_j, t_e\}$. For instance, one example in this dataset is:
r_i, r_j: `skirt, belt`;
c_i, c_j: `kilt, studded`;
t_e:`The outfit looks cohesive because the oversized layers are cinched with a studded belt, which complements the little strip from a kilt skirt that is also affixed to the belt, creating a visually pleasing balance in the outfit.`

We also report the percentages of each item in this dataset. For every item, we calculate the percentage of each item appearing in the whole dataset[3].

Explanations Generation In the domain of natural language processing, there is the general routine of training the model on a large corpus and then finetuning it on the domain-specific dataset (Thangarasa et al. 2023; Gupta et al. 2021). Thus we utilize a pre-trained language model to finetune it on our PFE-dataset. The idea is that we could create a template to construct the prompt as input to the language model, with the target sentences being the output.
Specifically, the objective of training is:

$$\max P_{g_\varphi}(t_e|\text{template}(r_i, r_j, c_i, c_j)) \tag{4}$$

We denote x_s as the generated word at step s, then the objective becomes:

$$\max \sum_{s=1}^{|t_e|} P_{g_\varphi}(x_s|\text{template}(r_i, r_j, c_i, c_j) + x_{:s}) \tag{5}$$

After training on the PFE-dataset, we input the features extracted from Stage I into the model g_φ to yield the explanations for the features in Stage I. The overall framework of our method is shown in Figure 2.

Instantiation of Models

Models for Stage I Various extraction models could serve as f_θ; we discuss two specific implementations:
Cross-Attn. We estimate the attention scores for each pair given t_i and t_j, then compute the weighted average of all concatenated pair embeddings. The resulting vector inputs into an MLP to produce the prediction. We put the implementation details in GitHub[2].
Rationale Extraction. Provided t_i and t_j and their respective binary labels, we utilize a rationale extraction approach to highlight the most significant words from each sentence. Specifically, we concatenate t_i and t_j into a single string t separated by the `<end>` token. Following the rationale extraction paradigm (Lei, Barzilay, and Jaakkola 2016; Bastings, Aziz, and Titov 2019; He et al. 2022), we train an extractor f_θ and a classifier h_ϕ. The trained f_θ extractor serves as the extraction model in our pipeline (Figure 2).

Models for Stage II Following the pretraining-finetuning routine in the state-of-the-art methods of explanation generation in recommendation (Li, Zhang, and Chen 2022; Li et al. 2023), we conduct experiments with the representative language models: **GPT-2**. a large transformer-based language model with 124 million parameters (Radford et al. 2019), can generate simple, albeit not necessarily comprehensive, sentences. GPT-2 serves as a basic baseline in our study. **Flan-T5-large**. Flan-T5 (Chung et al. 2022) is an augmented version of T5 (Raffel et al. 2020) and is trained on chain-of-thought data. The Flan-T5-large model is one of several models in the Flan-T5 series, boasting 780 million parameters. **Flan-T5-xl**. Also part of the Flan-T5 series, comprises 3 billion parameters.

[1]https://github.com/wangyu-ustc/PairFashionExplanation/tree/main#data-preprocessing-of-pfe-dataset

[2]https://github.com/wangyu-ustc/PairFashionExplanation/tree/main#cross-attention-extractor

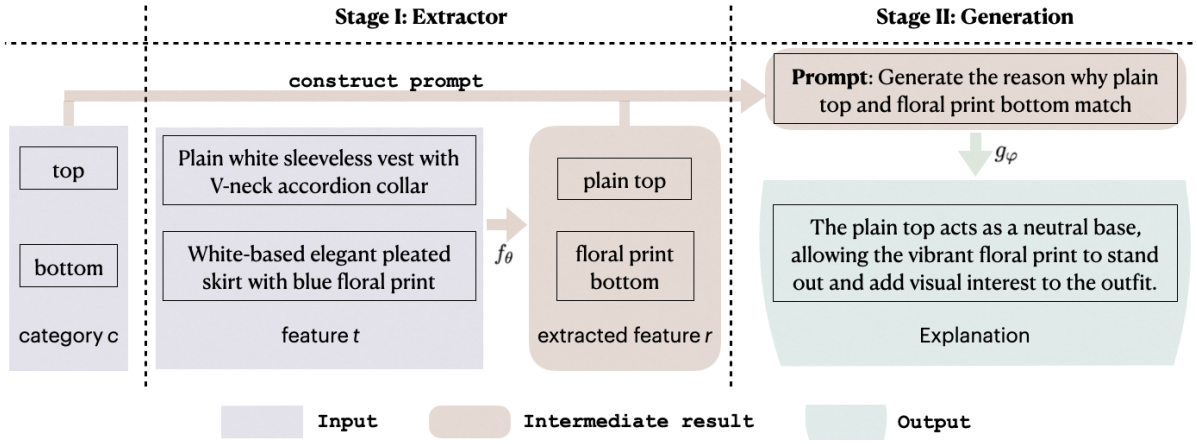


Figure 2: Overall Proposed Pipeline. The features t (in the form of bag of words) of two items are input into the extraction model f_θ (without the category) to obtain the extracted features r. Then we construct the prompts with the category c and the extracted features r, which are input into the generation model g_φ, yielding the final explanation.

Experiments

Experimental Setup

Datasets for Stage I To train models f_θ and h_ϕ in accordance with Eq.(3), we construct a composite dataset **Combined** from **Amazon Reviews**(Ni, Li, and McAuley 2019) and **FashionVC**(Song et al. 2017). The Amazon Review dataset comprises numerous subsets; we focus on the largest subset related to fashion, specifically "Clothing, Shoes and Jewelry". For each item, we leverage the "title" attribute and the "also_buy" list to build pairs (i, j), where i and j are individual items. We extract the categories c_i and c_j via manual noun dictionary matching, treating the remaining text as the features t_i and t_j used in Eq. (1). In this way, we force the model to determine if it is a good pair based on the features rather than the categories. By excluding pairs where both items belong to the same category, our dataset ultimately contains 124,679 pairs with 77,113 items. The FashionVC dataset includes 20,715 pairs of tops and bottoms, consisting of 14,871 tops and 13,663 bottoms. Here, each top or bottom possesses a corresponding category and feature, which we utilize as c_i and t_i for training. **Combined**. Recognizing the similar knowledge embedded within the two datasets, we integrate them into a single dataset, yielding a total of 145,394 pairs and 105,647 unique items. To generate negative samples, for each positive pair (i, j), we randomly select another item j', provided that (i, j') is absent from the positive pair set and i and j' belong to distinct categories. The final dataset includes 145,394 negative pairs.

Evaluation Metrics For automated assessment, following Li, Zhang, and Chen (2022), we use BLEU scores (Papineni et al. 2002) and Rouge scores (Lin 2004) to evaluate the quality of the generated text, BLEURT (Sellam, Das, and Parikh 2020) to compare the semantic similarity of our generated sentences to human-written sentences. Additionally, we use the Frechet Inception Distance (FID) (Heusel et al. 2017) to measure the consistency between the distribution of our model's output and the original sample distribution. We use CLIP (Radford et al. 2021) ViT-B/32 text encoder to encode the texts before calculating the distribution distance.

Implementation Details

Stage I Details We partition the Combined dataset into training, validation, and test sets at a ratio of 8:1:1. We jointly train the extractor f_θ and the classifier h_ϕ as depicted in Eq.(3) to acquire the extractor f_θ. For Cross-Attn, we apply lasso regularization with a weight value of 0.01. For Rationale Extraction, we fix the selection ratio at 0.3, indicating we select 30% of text from t_i and t_j as the rationale.

Stage II Details We employ prompts "f_i c_i and f_j c_j match because" and "Generate the reason why f_i c_i and f_j c_j match:" for GPT2 and Flan-T5, respectively. For GPT2, we set the batch size as 5 to train the whole model. While for Flan-T5-large and Flan-T5-xl, we use the package `peft` (Mangrulkar et al. 2022) with the method LoRA (Hu et al. 2022) to finetune. For Flan-T5-large, we set the batch size as 5. For Flan-T5-xl, we set the batch size as 1 but accumulate the gradient over 5 iterations. The learning rate for all models is set to 2e-4. The whole dataset is split into training set and testing set with the ratio of 9:1. The held-out testing set is also used for the evaluation in Table 1.

Overall Performance Comparison

We present a comprehensive comparison of our model's performance against several baseline models, which fall into the following groups:

Naive Repetition: This model merely repeats the input features, hence offering a simple benchmark.

Recommendation Models: Models such as PETER (Li, Zhang, and Chen 2021) and PEPLER (Li, Zhang, and Chen 2022), originally designed for explaining item recommendations, are repurposed for this task with slight modifications and finetuning on the PFE dataset.

Large Language Models: General-purpose language models like GPT2, Flan-T5-large, Flan-T5-xl, and ChatGPT, are employed without any specific fine-tuning on the PFE-dataset. For ChatGPT, we use the version `gpt-3.5-turbo` for comparison.
Ours: GPT2, Flan-T5-large, and Flan-T5-xl finetuned on the PFE-dataset training set.

Automatic Evaluation

Evaluation with PFE-dataset Our evaluation is predominantly based on the PFE dataset test set. For each data instance, the model generates an explanation whose distance from the ground truth explanation is measured and summarized. The results presented in Table 1 illustrate several key findings: (1) Our model, when finetuned on the PFE-dataset, exhibits superior performance in generating explanations that align closely with the ground truth. This underlines the value of training language models specifically for explanation generation tasks. (2) Despite recommendation models being domain-specific, their adaptations struggle with multi-object explanation generation. This underscores the complexities and unique aspects of explanation generation that necessitate a distinct, tailored approach. (3) Large language models, although capable of generating semantically coherent sentences and capturing entities from prompts (evident from high BLEU-1 scores), fall short in generating highly relevant explanations, as evidenced by their relatively lower BLEURT scores. This discrepancy highlights the pivotal role of training data like the PFE-dataset, which provides specific ground-truth compatibility explanations, thereby enhancing model performance in generating more accurate and meaningful explanations.

Evaluation for whole process Moreover, we evaluate our two-stage framework by comparing it with other models that lack an extraction stage. Here, the evaluation metric is the FID between generated explanations and the PFE-dataset. Table 2 demonstrates that our models yield significantly lower FID scores, signifying that our models are more adept at producing explanations that resemble the distribution of the ground truth dataset. The results suggest that our models are generating explanations, not merely describing the two items, which is crucial for the task at hand.

Human Evaluation Our human-centric evaluation involves comparing the entire framework with the top-performing models identified earlier. From the Combined dataset, we randomly selected 50 pairs and generated explanations using PEPLER, ChatGPT, and our fine-tuned Flan-T5-xl with the Rationale Extraction model. We posed the task as a rating problem for explanations, where each explanation was rated on a scale of 1-10. Three Amazon Turk workers, with a minimum `HIT Approval Rate` of 80%, independently rated each explanation. The average rating for each model is reported in Table 3. Our model consistently outperforms PEPLER and maintains competitive performance with ChatGPT, which is widely recognized for its capacity to generate fluent and coherent sentences. The fact that our model can match the performance of ChatGPT suggests that it is not only producing linguistically competent explanations but is also effectively capturing and conveying the complex relationship between the pair of items.

Ablation Study

Ablation Study for Stage I

Overall Performance of the Extraction Model The performance of the Cross-Attn and Rationale Extraction models are detailed in Table 4. Both models display an ability to discern compatibility correlations, with the Rationale Extraction model performing slightly better. This could be due to its more complex LSTM structure.

Influence of Extraction Models on Explanation Generation Fidelity We investigated the effect of the extraction models on the overall quality of generated explanations by calculating the FID between the CLIP (Radford et al. 2021) embeddings of the generated explanations and the ground-truth explanations. We adopt CLIP ViT-L/14 for the calculation. The results are presented in Table 5. Two key observations can be made from the results: (1) All models equipped with extraction methods were capable of producing explanations more closely resembling ground-truth explanations. (2) Interestingly, the finetuned GPT2 model generated explanations that have the smallest FID. Although this only suggests that GPT2 can better mimic the pattern of the training corpus rather than identifying compatibility relationships, it nevertheless implies potential utility for future studies.

Ablation Study for Stage II

Human Evaluation of Generation Model with Stage I Keywords Upon the successful training of Stage I and Stage II, we conduct the generation with the models f_θ and g_ϕ by inputting the extracted features (with f_θ) of true positive pairs into the Stage II generation model (g_ϕ). The rationale extraction model served as our primary extractor.
We also engaged ChatGPT in our experiment. For each item pair characterized by features f_i, f_j and categories c_i, c_j, we queried ChatGPT with the following question:
`What is the reason why` f_i c_i `and` f_j c_j `match, and could you please provide a concise response (with one or two sentences) that can be directly shown to customers?`
Post explanation generation from different models, we carried out a user study as detailed below:

Evaluation of Generation : We randomly selected 50 prompts from the first stage and leveraged GPT2, Flan-T5-large, Flan-T5-xl (all finetuned), and ChatGPT to generate explanations. Subsequently, for each item pair, we asked the users to score the explanations on a 1-10 scale, focusing on their conciseness, persuasiveness, and overall acceptability, which would convince them that the provided pair of items indeed form an excellent match. We delegated each question to three Amazon Turk workers, maintaining a criteria that their `HIT Approval Rate` must exceed 80%. Table 6 summarizes the results. From this table, two observations can be made: (1) Flan-T5-xl outperformed ChatGPT when the extracted features from Stage I were used, showcasing

Model	BLEURT	BLEU-1	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	FID
Naive Rep	-0.3731	27.35	5.78	0.3963	0.1391	0.2849	8.670
PETER	-1.2367	0.02	0.00	0.0005	0.0000	0.0005	60.004
PEPLER-F	-0.3115	31.52	3.76	0.2892	0.0586	0.2327	10.313
GPT-2	-0.6656	10.47	0.17	0.1039	0.0051	0.0822	33.847
Flan-T5-large	-0.6398	15.33	3.16	0.3194	0.1154	0.2349	9.047
Flan-T5-xl	-0.5783	23.64	4.37	0.2825	0.1004	0.2179	9.634
ChatGPT	0.0139	35.67	6.34	0.3872	0.1274	0.2739	5.802
GPT-2-F	-0.0853	38.18	9.01	0.3868	0.1442	0.2990	3.014
Flan-T5-large-F	0.0413	43.80	12.57	0.4557	0.2081	**0.3538**	**2.256**
Flan-T5-xl-F	**0.0736**	**46.12**	**14.11**	**0.4691**	**0.2208**	0.3485	3.046

Table 1: Overall Performance Comparison on the held-out test set of PFE dataset, measuring the alignment between the generated explanations and the ground-truth compatibility relationships. "Naive Rep" means "Naive repetition". "F" means being fine-tuned with the PFE dataset.

	Naive Rep	PEPLER	GPT2-RE	FT5-large-RE	FT5-xl-RE	ChatGPT	GPT2-F-RE	FT5-large-F-RE	FT5-xl-F-RE
FID	20.657	14.918	40.423	20.086	23.951	18.667	7.689	8.091	11.657

Table 2: Comparison of FID for various models. "F" refers to being fine-tuned, "RE" means the model is equipped with the Rationale Extraction (RE) component as the extractor. "FT5" means Flan-T5.

Model	PEPLER	ChatGPT	RE + Flan-T5-xl-F
Avg Rating	5.48	**6.50**	6.43

Table 3: Human Evaluation.

Method	Acc	R@5	R@10	R@20	R@50
Cross-Attn	0.8505	0.1468	0.2229	0.3187	0.4846
RE	0.8531	0.3060	0.3984	0.5049	0.6596

Table 4: Performance of different models in Stage I.

	None	Cross-Attn	Rationale Extraction
GPT2-F	7.980	**6.718**	7.689
Flan-T5-large-F	11.141	**8.073**	8.091
Flan-T5-xl-F	16.647	**10.518**	11.657

Table 5: FID with different extraction models.

Model	GPT2	Flan-T5-large	Flan-T5-xl	ChatGPT
Avg Rating	5.91	6.27	**6.51**	6.37

Table 6: User study for different generation models.

our model's ability to generate more user-acceptable explanations. (2) Flan-T5-xl, compared with Flan-T5-large and GPT2, recorded the best performance, indicating that larger models possess superior abilities to internalize knowledge from the finetuned dataset.

Influence of Training Sample Size To investigate whether the PFE-dataset, consisting of 6,405 examples, is sufficient for the model to discern the compatibility relationships between the paired items, we performed experiments by finetuning the LLM with a subset of the PFE-dataset. The results are illustrated in Figure 3. From this figure, it can be concluded that: (1) Larger models necessitate fewer samples for finetuning. The smallest model, GPT2, exhibits enhanced performance with an increasing number of samples, while the gains for Flan-T5-large are relatively modest when the sample size increases. As for Flan-T5-xl, the BLEURT and Rouge-2 scores barely rise when samples are increased from 5,000 to 6,000. (2) Although 6,405 samples (5,764 samples used for training) might appear to be limited, the results suggest that further sample augmentation may not significantly boost the performance.

Case Study

Illustrative Examples after Combining Stage I and Stage II We show generated explanations from PEPLER, ChatGPT, and our Flan-T5-xl model in Table 7. In the example, we find that the recommendation model PEPLER may fall short in terms of the fluency of the explanations. Both ChatGPT and Flan-T5-xl capture the concept of "casual", showing that our model can compete favorably with ChatGPT. This aligns with the quantitative results in Table 3 where the rating of our model is similar to the rating of ChatGPT.

Illustrative Examples from Stage II In Stage II, we present generated examples in Table 8. These are selected from the test set of the PFE-dataset, with additional results available in Github[3]. From the table, we observe that our

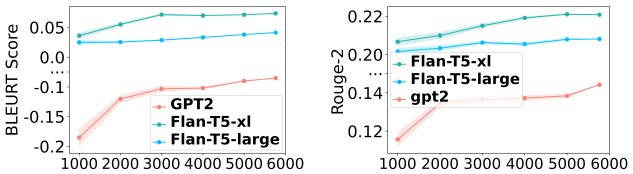
[3]https://github.com/wangyu-ustc/PairFashionExplanation/tree/main?tab=readme-ov-file#additional-experiments

Item/Feature	boot/timberland men groveton ltt chukka; shirt/ularmo men boy summer cotton tees short
PEPLER-F	the black color of the shirt complements the black color of the pants, creating a cohesive color scheme. the black color of the shirt also ties the whole outfit together.
ChatGPT	The Timberland men's Groveton LTT Chukka Boot and Ularmo Men Boy Summer Cotton Tees Short Star Printed T L Black Shirt Sleeve Shirt match because the casual yet stylish design of the boots complements the graphic print of the t-shirt.
Flan-T5-xl-F-RE	The Timberland ltt chukka boot and ularmo summer cotton tees shirt sleeve shirt complement each other because they both have a casual and laid-back vibe, creating an effortless outfit.

Table 7: Our model Flan-T5-xl is equipped with the Rational Extraction model from the first stage, and "F" means fine-tuned with PFE-dataset. ChatGPT and PEPLER-F take the raw features shown in the first row as input.

Item/Feature	Jacket and mini skirt/Matching black tweed cropped; heels/vibrant pink with oversized black bows on the back
GPT2-F	The white tee with oversized black bows on the back match the black tweed cropped jacket, and the vibrant pink heels add pops of contrasting color to tie the whole outfit together.
Flan-T5-large-F	The black tweed jacket and mini skirt match each other in terms of color and texture, while the vibrant pink heels with oversized black bows on the back add a pop of color that complements the black and pink tones in the outfit. Overall, the combination creates a stylish look.
Flan-T5-xl-F	The black tweed cropped jacket and mini skirt create a cohesive look, while the vibrant pink heels with oversized black bows on the back add a pop of color to the outfit.
ChatGPT	The reason why a matching black tweed cropped jacket and miniskirt pair well with vibrant pink heels featuring oversized black bows on the back is that the combination creates a bold and stylish contrast, combining classic elegance with playful accents.
Ground Truth	The black tweed fabric of the jacket and mini skirt creates a cohesive outfit, while the vibrant pink slingback heels with the oversized black bows add a pop of color and playful touch to the overall look.

Table 8: Generation results in testset of PFE dataset of Stage II. "F" means fine-tuned with PFE-dataset.



(a) BLEURT (b) Rouge-2

Figure 3: Effect of the number of sentences. The x-axis is the number of sentences used for training.

models Flan-T5-large-F and Flan-T5-xl-F mention the crucial point that the pink heels add "a pop of color", a detail that ChatGPT fails to touch upon. This shows that although ChatGPT may excel at sentence construction, it might miss capturing important compatibility relations between items, and our model is more sensitive to the domain-specific relations compared with general-purpose LLMs.

Conclusion

In this paper, we address the challenge of generating explanations for the compatibility of paired fashion items, an area that limited research focuses on. The value of our work is twofold. First, we construct the PFE-dataset that can be leveraged for further research in Fashion Recommendation. Secondly, our framework offers a generalized approach for revealing compatibility relationships, extending its potential application beyond the confines of our initial focus on fashion items. Notably, our framework is not limited to the realm of fashion; it also has potential applications in determining the compatibility of various pieces of furniture for interior design purposes and establishing the interoperability of different electronic devices. A current limitation of our method is that it may find multi-object compatibility relationships challenging to reveal. Thus in the future, we aim to extend our framework into explaining multi-object compatibility relationships. The applications of our approach include curating collections of fashion items or coordinating comprehensive interior designs. In conclusion, our work serves as a stepping stone towards a better understanding of the underlying factors that contribute to the compatibility of items, and offers a versatile tool that holds promise for wider application and further refinement.

References

Balim, C.; and Özkan, K. 2023. Diagnosing fashion outfit compatibility with deep learning techniques. *Expert Systems with Applications*, 215: 119305.

Banerjee, D.; Rao, K. S.; Sural, S.; and Ganguly, N. 2020. BOXREC: Recommending a Box of Preferred Outfits in Online Shopping. *ACM Trans. Intell. Syst. Technol.*, 11(6): 69:1–69:28.

Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Bastings, J.; Aziz, W.; and Titov, I. 2019. Interpretable Neural Predictions with Differentiable Binary Variables. In *ACL (1)*, 2963–2977. Association for Computational Linguistics.

Castro, E.; Ferreira, P. M.; Rebelo, A.; Rio-Torto, I.; Capozzi, L.; Ferreira, M. F.; Gonçalves, T.; Albuquerque, T.; Silva, W.; Afonso, C.; Sousa, R. G.; Cimarelli, C.; Daoudi, N.; Moreira, G.; Yang, H.; Hrga, I.; Ahmad, J.; Keswani, M.; and Beco, S. C. 2023. Fill in the blank for fashion complementary outfit product Retrieval: VISUM summer school competition. *Mach. Vis. Appl.*, 34(1): 16.

Chen, W.; Huang, P.; Xu, J.; Guo, X.; Guo, C.; Sun, F.; Li, C.; Pfadler, A.; Zhao, H.; and Zhao, B. 2019. POG: personalized outfit generation for fashion recommendation at Alibaba iFashion. In *SIGKDD*, 2662–2670.

Chen, Y.; Zhou, Z.; Lin, G.; Chen, X.; and Su, Z. 2022. Personalized Outfit Compatibility Prediction based on Regional Attention. In *2022 9th International Conference on Digital Home (ICDH)*, 75–80. IEEE.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Guan, W.; Song, X.; Zhang, H.; Liu, M.; Yeh, C.; and Chang, X. 2022. Bi-directional Heterogeneous Graph Hashing towards Efficient Outfit Recommendation. In *ACM Multimedia*, 268–276. ACM.

Gupta, T.; Zaki, M.; Krishnan, N. M. A.; and Mausam. 2021. MatSciBERT: A Materials Domain Language Model for Text Mining and Information Extraction. *CoRR*, abs/2109.15290.

He, Z.; Wang, Y.; McAuley, J. J.; and Majumder, B. P. 2022. Controlling Bias Exposure for Fair Interpretable Predictions. In *EMNLP (Findings)*, 5854–5866. Association for Computational Linguistics.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NIPS*, 6626–6637.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*. OpenReview.net.

Kaicheng, P.; Xingxing, Z.; and Wong, W. K. 2021. Modeling Fashion Compatibility with Explanation by using Bidirectional LSTM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3894–3898.

Lei, T.; Barzilay, R.; and Jaakkola, T. S. 2016. Rationalizing Neural Predictions. In *EMNLP*, 107–117. The Association for Computational Linguistics.

Li, J.; He, Z.; Shang, J.; and McAuley, J. 2023. Unifying aspect planning and lexical constraints for generating explanations in recommendation. In *KDD*.

Li, K.; Liu, C.; and Forsyth, D. 2019. Coherent and controllable outfit generation. *arXiv preprint arXiv:1906.07273*.

Li, L.; Zhang, Y.; and Chen, L. 2021. Personalized transformer for explainable recommendation. *arXiv preprint arXiv:2105.11601*.

Li, L.; Zhang, Y.; and Chen, L. 2022. Personalized prompt learning for explainable recommendation. *arXiv preprint arXiv:2202.07371*.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Lin, Y.; Moosaei, M.; and Yang, H. 2020. OutfitNet: Fashion Outfit Recommendation with Attention-Based Multiple Instance Learning. In *WWW*, 77–87. ACM / IW3C2.

Lin, Y.; Ren, P.; Chen, Z.; Ren, Z.; Ma, J.; and De Rijke, M. 2019. Explainable outfit recommendation with joint outfit matching and comment generation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8): 1502–1516.

Lu, S.; Zhu, X.; Wu, Y.; Wan, X.; and Gao, F. 2021a. Outfit compatibility prediction with multi-layered feature fusion network. *Pattern Recognition Letters*, 147: 150–156.

Lu, Z.; Hu, Y.; Chen, Y.; and Zeng, B. 2021b. Personalized Outfit Recommendation With Learnable Anchors. In *CVPR*, 12722–12731. Computer Vision Foundation / IEEE.

Mangrulkar, S.; Gugger, S.; Debut, L.; and Younes Belkada, S. P. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. https://github.com/huggingface/peft.

Mo, D.; Zou, X.; Pang, K.; and Wong, W. K. 2023. Towards private stylists via personalized compatibility learning. *Expert Systems with Applications*, 219: 119632.

Moosaei, M.; Lin, Y.; Akhazhanov, A.; Chen, H.; Wang, F.; and Yang, H. 2022. OutfitGAN: Learning Compatible Items for Generative Fashion Outfits. In *CVPR Workshops*, 2272–2276. IEEE.

Ni, J.; Li, J.; and McAuley, J. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 188–197.

OpenAI. 2023. GPT-4 technical report. https://arxiv.org/abs/2303.08774. Accessed: 2024-02-28.

Papadopoulos, S.; Koutlis, C.; Papadopoulos, S.; and Kompatsiaris, I. 2022. VICTOR: Visual Incompatibility Detection with Transformers and Fashion-specific contrastive pretraining. *CoRR*, abs/2207.13458.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67.

Sarkar, R.; Bodla, N.; Vasileva, M. I.; Lin, Y.; Beniwal, A.; Lu, A.; and Medioni, G. 2023. OutfitTransformer: Learning Outfit Representations for Fashion Recommendation. In *WACV*, 3590–3598. IEEE.

Sellam, T.; Das, D.; and Parikh, A. P. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *ACL*, 7881–7892. Association for Computational Linguistics.

Song, X.; Fang, S.; Chen, X.; Wei, Y.; Zhao, Z.; and Nie, L. 2023. Modality-Oriented Graph Learning Toward Outfit Compatibility Modeling. *IEEE Trans. Multim.*, 25: 856–867.

Song, X.; Feng, F.; Liu, J.; Li, Z.; Nie, L.; and Ma, J. 2017. Neurostylist: Neural compatibility modeling for clothing matching. In *Proceedings of the 25th ACM international conference on Multimedia*, 753–761.

Tangseng, P.; and Okatani, T. 2020. Toward explainable fashion recommendation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2153–2162.

Thangarasa, V.; Gupta, A.; Marshall, W.; Li, T.; Leong, K.; DeCoste, D.; Lie, S.; and Saxena, S. 2023. SPDF: Sparse Pre-training and Dense Fine-tuning for Large Language Models. *CoRR*, abs/2303.10464.

Wang, R.; Wang, J.; and Su, Z. 2022. Learning compatibility knowledge for outfit recommendation with complementary clothing matching. *Comput. Commun.*, 181: 320–328.

Zhou, Z.; Su, Z.; and Wang, R. 2022. Attribute-aware heterogeneous graph network for fashion compatibility prediction. *Neurocomputing*, 495: 62–74.